CONTEMPORARY CLASSIFIERS USED IN MAP REDUCED ALGORITHMS IN BIG DATA

Degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Sampada Thakkar(141335)
Rishika Bajaj(141272)

Under the supervision of

Dr. Hemraj Saini
Ms. Geetanjali

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Certificate

Candidate’s Declaration

I hereby declare that the work presented in this report entitled **“Contemporary classifiers in map reduced algorithm in big data”** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2016 to December 2016 under the supervision of **Dr. Hemraj Saini** (Assistant Professor,CSE).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sampada Thakkar(141335) Rishika Bajaj(141272)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)
Dr. Hemraj Saini
Assistant Professor
CSE
Dated:24/11/2017

Acknowledgement

I would like to use this opportunity to express my gratitude to everyone who supported me throughout the course of this B.Tech project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and inspiring views on a number of issues related to the project.

I am especially grateful to **Dr. Hemraj Saini**, Project Supervisor, for his valuable suggestions, support and constant encouragement during the course of the project. Hisperpetual energy, motivation, enthusiasm and immense knowledge inspired me to discipline myself in efficiently executing my multiple responsibilities simultaneously.

Date: 25/11/2017

Sampada Thakkar(141335)
Rishika Bajaj(141272)

Table of Content

List of Abbreviations

1. KNN- K Nearest Neighbor

2. AWS- Amazon Web Services

3. HDFS- Hadoop Distributed File System

4. SVM- Support Vector Machine

5. MLC – Maximum Likelihood Calculation

List of Figures

List of Graphs

List of Tables

Abstract

The term “Big Data” refers to large and complex data sets made up of a variety of structured and unstructured data which are too big, too fast, or too hard to be managed by traditional techniques.Structured: DBMS tables, CSV's and XLS's.Unstructured: e-mail attachments, manuals, images, PDF documents, medical records such as x-rays, ECG and MRI images, forms, rich media like graphics, video and audio, contacts, forms and documents.

Programming model for processing large-scale datasets in computer clusters. The Map Reduce programming model consists of two functions, map () and reduce ().The map () function takes an input key/value pair and produces a list of intermediate key/value pairs. The Map Reduce runtime system groups together all intermediate pairs based on the intermediate keys and passes them to reduce() function for producing the final results.

A classifier is a function f that maps input feature vectors $x \in X$ to output class labels $y \in \{1, . . . , C\}$, where X is the feature space. We will typically assume X = IRD or $X = \{0, 1\}$ D, i.e., that the feature vector is a vector of D real numbers or D binary bits, but in general, we may mix discrete and continuous features. We assume the class labels are unordered (categorical) and mutually exclusive.

CHAPTER-1

1.1 INTRODUCTION

Big Data :

Big Data is nothing but large volume of data. This data can be of any form. The emails we write, texts, documents, pictures and various amount and type of data. Data can be structured, semi structured and unstructured.

Structured Data:

It is the organized form of data and can best stored in database SQL. They are very simple to manage in rows and columns but only very small amount of data is unstructured formwhichisaround8-10%.

Semi-Structured Data:

It is not as organized as structured data but still has some organized data properties that makes it easier to analyze but it is also in very small amount which is around 8-10%. It cannot be organized in relational database. Examples are XML,JSON.

Unstructured Data: It comprises of 80% of data which consists of email, texts, multimedia messages, web pages. It cannot be organized in database. The real problem is not the increasing amount of data but organizing the data by reducing costs and time. All we need is to organize data according to the requirement. Following are the various measures according to which big data is characterized. Deep analysis of the data is done to characterize these data and show desired result. Hadoop is the solution of big data. It takes data and helps in characterizing this data.

Characterizing Big Data:

Volume:

Quantity of data which is generated or stored. This help us to know how much data is to be processed.

Variety :

Nature or type of data. This help in analyzing the data.

Velocity :

Speed of generation and processing of data.

Variability :

Hampering to handle and manage inconsistent of data set.

Veracity :

Varying data quality affecting analysis of accuracy.

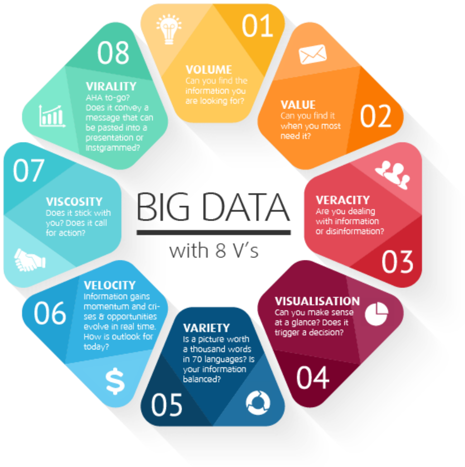

BIG DATA
with 8 V's
01
VOLUME
Can you find the information you are looking for?
02
VALUE
Can you find it when you most need it?
03
VERACITY
Are you dealing with information or disinformation?
04
VISUALISATION
Can you make sense at a glance? Does it trigger a decision?
05
VARIETY
Is a picture worth a thousand words in 70 languages? Is your information balanced?
06
VELOCITY
Information gains momentum and cri-ses & opportunities evolve in real time. How is outlook for today?
$
07
VISCOSITY
Does it stick with you? Does it call for action?
08
VIRALITY
AHA to-go? Does it convey a message that can be pasted into a presentation or Instgrammed?

Figure 1.1

Technology

- Amazon.com has a large number of back-end operations every minute. Amazon is based on Linux and it has large capacities of 7.8 TB and 24.7 TB.
- eBay.com uses two data warehouses at 7.5 peta bytes and 40PB as well as a 40PB Hadoop cluster for search, consumer recommendations, and merchandising.
- Facebook takes into account 50 billion photos from its user base.
- Google handles 100 billion searches per month in records.
- Oracle NoSQL database has been tested to past the 1M ops/sec mark with 8 shards and proceeded to hit 1.2M ops/sec with 10 shards.

Sampling Big Data:

Now, the point is if all the data is useful or not? The answer is No. We have to sample the data which means to looks for the right points where the useful data resides , We should know what we are aiming at and then sample the data accordingly. It is important to analyze the sentiment of each topic . It can be broken down to small units and then analyzed bit by bit.

Hadoop:

It is nothing but a framework which is based on java programming. Hadoop makes it easier for applications to run and supports big data. Hadoop is basically the solution for Big Data. Doug

Cutting and Mike created Hadoop in 2006. They took the inspiration from Map reduce of Google. Hadoop is continuously and constantly executed and updated . It is scalable highly on platforms of cloud like Microsoft Azure or Amazon Web Services(AWS). Hadoop kernel is used by Hadoop to provide essential libraries . HDFS also known as Hadoop Distributed File System stores very large amount of data while it aims at achieving high bandwidth between two nodes.

Hadoop YARN (Yet Another Resource Negotiator) aims to manage and schedule for a user applicant. Hadoop Map Reduce maps and then reduces to reach to the result.

There are many packages supported by Hadoop:

- Flume
- Hive
- Zookeeper
- Hbase
- Pig
- Sqoop
- Spark
- Phoenix
- Oozle

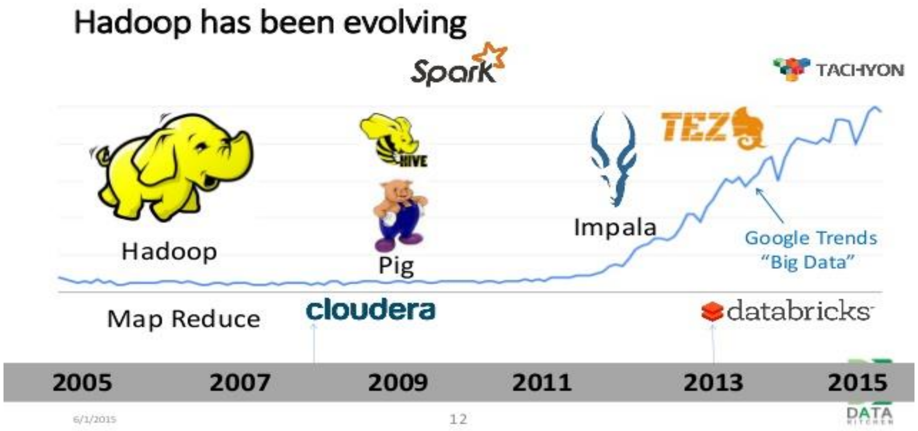


Figure 1.2

Map Reduce:

It is a model of programming which processes and generates a large amount of dataon a cluster.

It consists of map() and reduce(). Map() involves mapping and sorting of data while Reduce() does the exact opposite of mapping.It runs on split-apply-combine combination.

"Map" step:

When map() function is applied , output is written to a temporary storage . Redundant copy of input data is processed which is ensured by master node.

"Shuffle" step:

Data is redistributed on the basis of output keys which is done by worker nodes such that all data belongs to same worker node to one key.

"Reduce" step:

Each group of output data is processed by worker nodes in parallel.

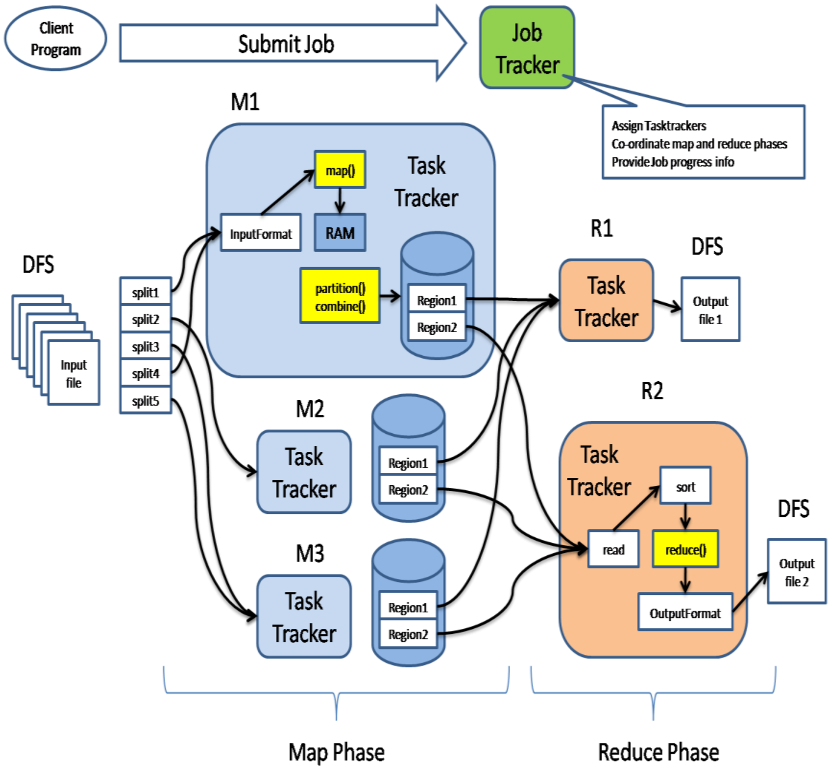


Figure 1.3

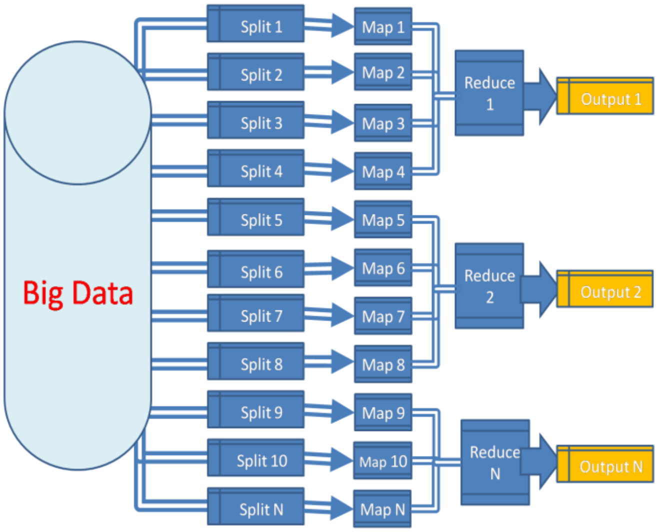


Figure 1.4

Classifiers

It only provides the basis of classification. Many techniques are being used to classify the data.It is a function “f” which is used to map vectors x belonging to X to output class labelled as y belonging to {1,2,...C},X being feature space, assuming labels of class are not ordered and exclusive mutually.

Challenges faced by Map Reduce Algorithms like Data Processing,Data Storage,Data Analysis and Security are all solved by Hadoop Architecture Layers. So to solve all these problems classifiers came into play.

Classifiers such as

ZeroR

OneR

Naïve Bayes

Decision Tree

Linear Regression

K nearest Neighbour

Logistic Regression

SVM

1.2 Problem Statement

1. Determining different type of classifiers available to classify big data using MapReduce,

2.Classification of different type of classifiers available.

3.Comparison between different type of classifiers and studying on the different basis.

4.Creating a new classifier by minimizing cost and time.

1.3Objective

The objective of the project is to classify different type of classifiers and compare them according to speed of execution and various differences. We aim to find or create a new classifier which take minimum time and space. The main goal is to implement map reduced algorithms to find the desired results.

CHAPTER-2

LITERATURE REVIEW

S.No	Author	Year	Title	Remarks
1.	M. Dhavapriya, N. Yasodha	2016	Big Data Analytics: Challenges and Solutions Using Hadoop, Map Reduce and Big Table	It gives an idea about big data and hadoop and its architecture layers . The working of tools like map reduce ,hive, pig , spark , flume.
2.	Katarina Grolinger1 , Michael Hayes1 , Wilson A. Higashino1,2, Alexandra L'Heureux1 David S. Allison1,3,4,5, Miriam A.M. Capretz1	2014	Challenges for Map Reduce in Big Data	This paper is all about the challenges faced by using map reduce algorithms like data storage , data analytics, online data processing and security of data.
3.	Kevin P.Murphy	2006	Naive Bayes classifiers	Introduction of naïve bayes classifiers and working of predictors of algorithms .

Big Data is a large amount of structured and unstructured data. Hadoop and Hdfs are the methods of processing and storing data

Challenges faced by Map Reduce Algorithms like Data Processing,Data Storage ,Data Analysis and Security are all solved by Hadoop Architecture Layers

Classifiers such as:

-ZeroR

-OneR

-Naïve Bayes

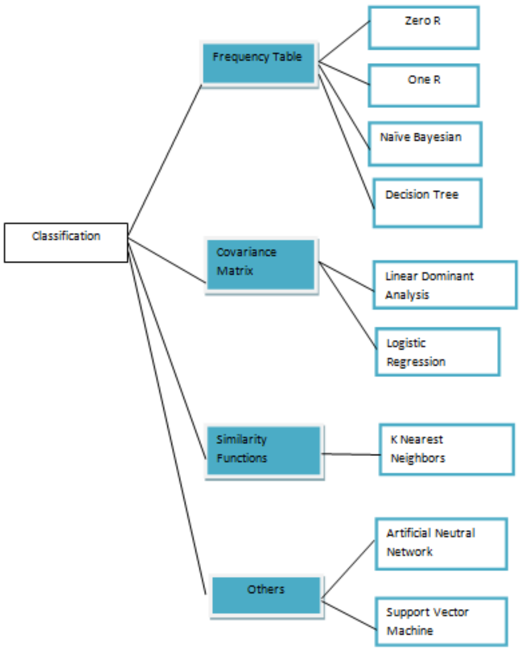
-Decision Tree

-Logistic Regression

-Linear Discriminant Analysis

-K Nearest Neighbors

And their working.



2.1 Zero R:

ZeroR is based on frequency table. ZeroR stands for zero which means it lays its focus on the target and ignores all predictors.

ZeroR classifier stresses on predicting the majority category (class).

Although ZeroR has no predictability power, it is useful for determining a baseline performance as a benchmark for other classification methods.

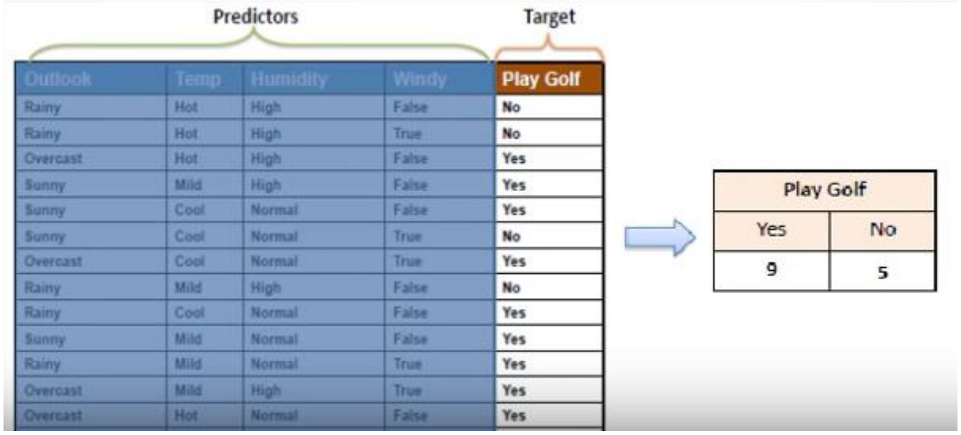
By baseline we mean that this is the least accurate classifier that we can use. This means if we develop a model and its accuracy is worse than this then the model is useless.

Working of ZeroR:

We construct a frequency table for the target and select the value that appears the most in it. That means we ignore the other features and only look at the class for building the frequency table and for any new input, we always predict it to be as the majority of the classes from the class column.

ZeroR Example:

"Play Golf=YES" is the ZeroR model for the following dataset which has the accuracy of 0.64



Predictors				Target
Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

Play Golf	
Yes	No
9	5

Here we have four predictors and the fifth column is the class. What ZeroR does is that it ignores all the predictors and it only builds a frequency table based on the class. If we have more than one class, we count it's frequency as well.

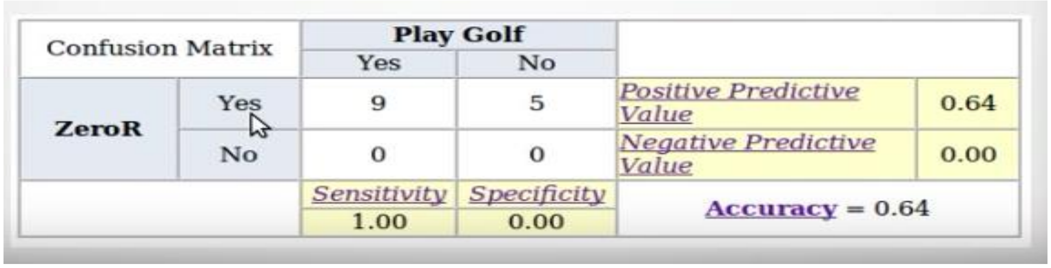
Here we have nine “yes” and five “no”, we have fourteen instances or observations as shown in the above figure. Now any future input would be predicted to be of type “yes” because it is the majority class. Any new input for example of outlook-‘Rainy’, temperature-‘hot’, humidity-‘normal’, windy-‘true’ and we want to guess whether to play or not then the ZeroR will always guess it to be a “yes” because that’s the majority class.

Now from the above frequency table we can easily build a confusion matrix to evaluate the performance.

ZeroR Model Evaluation:

The Confusion Matrix which is shown below depicts that ZeroR predicts the majority class only correctly.

As mentioned before, ZeroR is only useful for determining a baseline performance for other classification methods .



Confusion Matrix		Play Golf			
		Yes	No		
ZeroR	Yes	9	5	Positive Predictive Value	0.64
	No	0	0	Negative Predictive Value	0.00
		Sensitivity	Specificity	Accuracy = 0.64	
		1.00	0.00		

Figure 2.1

Now for our classifier because we predict everything to be a “yes”, here in the confusion matrix we have the count of the actual classes in the horizontal and the count for the predicted classes in the vertical. We have nine “yes” because we have fourteen points now and all of them are classified to be “yes”. So then we have nine as our true positive (actually “yes” predicted to be “yes”) and we have five as false positive(actually “no” but predicted to be “yes”).

So we take out Positive Predictive Value, Negative Predictive Value, Sensitivity and Specificity and finally calculate the accuracy from these values.

2.2 OneR:

Unlike ZeroR classifier which ignores the predictors in the data and it only builds a frequency table based on the class column and it predicts everything to be same as the majority class, the OneR classifier is another classifier based on the frequency table.

OneR, also known as “One Rule”, is a simple and accurate classification algorithm. So instead of ignoring all the predictors, it only chooses one predictor and it uses it for classification.

How It Works:

One rule for each predictor is generated in the data, then selects the rule with the smallest total error as its “one rule”.

For creating the predictor rule, a frequency table has to be constructed against target for each and every predictor.

It is evidently shown that only slightly less accurate rules are produced by OneR than the other classification algorithms while producing rules that are simple to be interpreted by the human beings.

Algorithm:

```
For each predictor;
    For each value of the predictor, make a rule as follows:
    Count how often each value of target (class) appears.
    Find the most frequency class
    Make the rule assign that class to this value of the predictor.
    Calculate the total error of the rules of each predictor.
    Choose the predictor with the smallest total error.
```

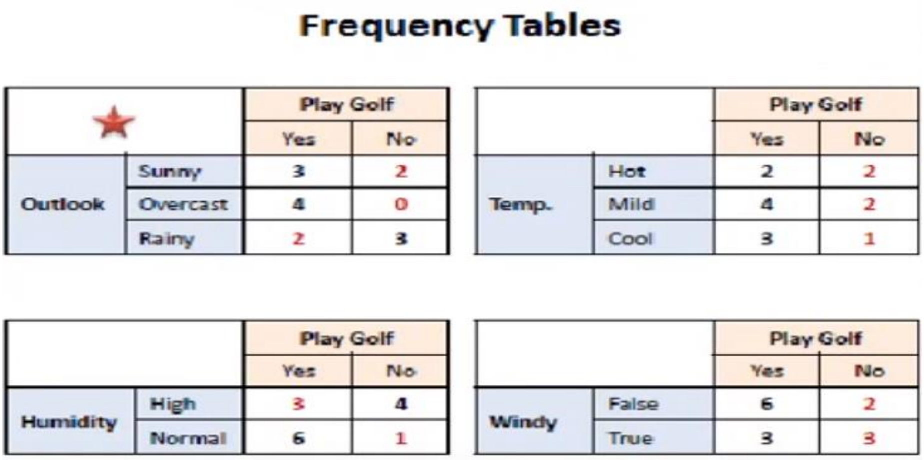
Example:

Now taking the previous example:

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

Figure 2.2

There are four predictors in the above example and we build a frequency table based on each predictor against the class as shown below:



Frequency Tables

★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

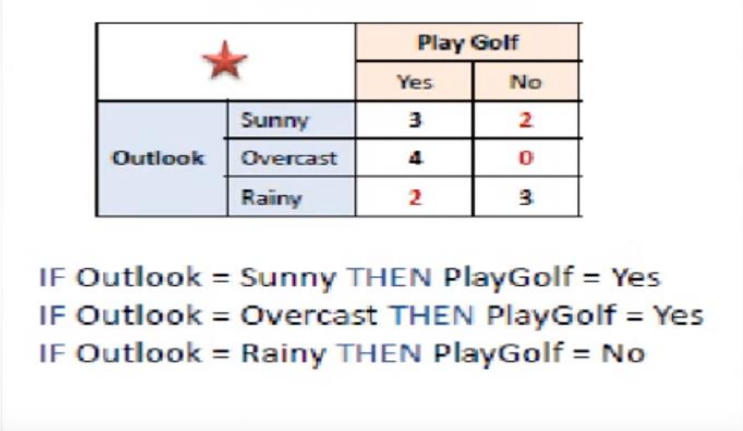
		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Figure 2.3

We can see in the frequency tables that we have different categories for different attribute predictors. The sum of all the number of "yes" and "no" should be same as the number of instances. If they don't then something is not right. From these we can built a confusion matrix to calculate the accuracy or to somehow find a way of measuring the error. After building a confusion matrix we will find that the best predictor is ‘outlook’ because it gives us the highest accuracy. The rules are given below:



★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

IF Outlook = Sunny THEN PlayGolf = Yes
IF Outlook = Overcast THEN PlayGolf = Yes
IF Outlook = Rainy THEN PlayGolf = No

Figure 2.4

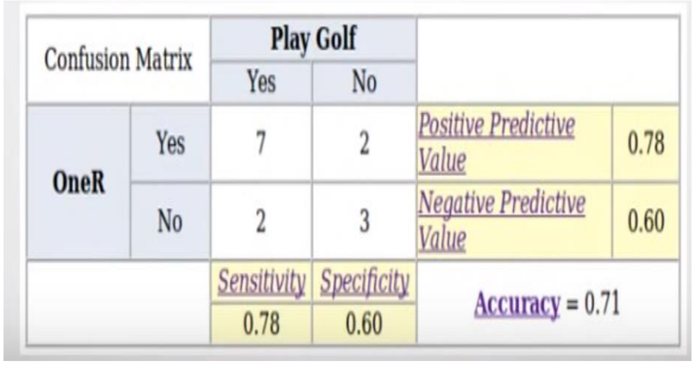
Given above are the three rules for the best predictor. Here we find the number of times our target class appears . For example, if outlook is sunny then we always choose “yes”, likewise if its overcast then we always choose “yes” and if rainy, we always choose “no”.

Predictors Contribution:

Evidently, the total error which is calculated from the frequency tables shows the measure of contribution of each predictor.

A low total error implies that the contribution is higher to the probability of the model.

Confusion matrix for outlook:



Confusion Matrix		Play Golf			
		Yes	No		
OneR	Yes	7	2	Positive Predictive Value	0.78
	No	2	3	Negative Predictive Value	0.60
		Sensitivity	Specificity	Accuracy = 0.71	
		0.78	0.60		

Figure 2.5

Here we see the number of true positives, true negatives, false positives and false negatives and then compute the accuracy. We can do this for each attribute and choose the one with the highest accuracy, in this case it will be outlook as it gives 71% accuracy.

Naïve Bayesian:

This classifier is another classifier based on the frequency table.

The Naive Bayesian classifier is based on Bayes' theorem with the assumption of independence between predictors which means that knowing the value of one attribute does not tell us anything about the another attribute or the another predictor.

This model of Naïve Bayesian can be bulid easily, without any complicated iterative parameter estimation which can be used to make it extremely useful for datasets that are large.

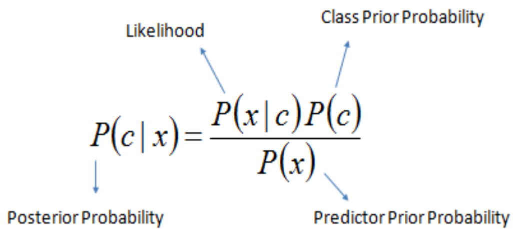
The Naïve Bayesian classifier is very simple and yet it does surprisingly well and is widely used because of its outstanding performance over more sophisticated classification methods.

How It Works:

A simple way of calculating the posterior probability is provided by Bayes theorem, P(c|x), from P(c), and P(x|c).

Naïve Bayes classifier is based on the assumption that the change in the value of a predictor (x) has no effect on a given class (c), stating that it is independent of the values of other predictors.

This assumption is called class conditional independence.



$$P(c\,|\,\mathrm{X}) = P(x_1\,|\,c)\times P(x_2\,|\,c)\times\cdots\times P(x_n\,|\,c)\times P(c)$$

P(c|x) is the posterior probability of class (target) given predictor(attribute). P(c) is the prior probability of class. P(x|c) is the likelihood which is the probability of predictor given class. P(x) is the prior probability of predictor.

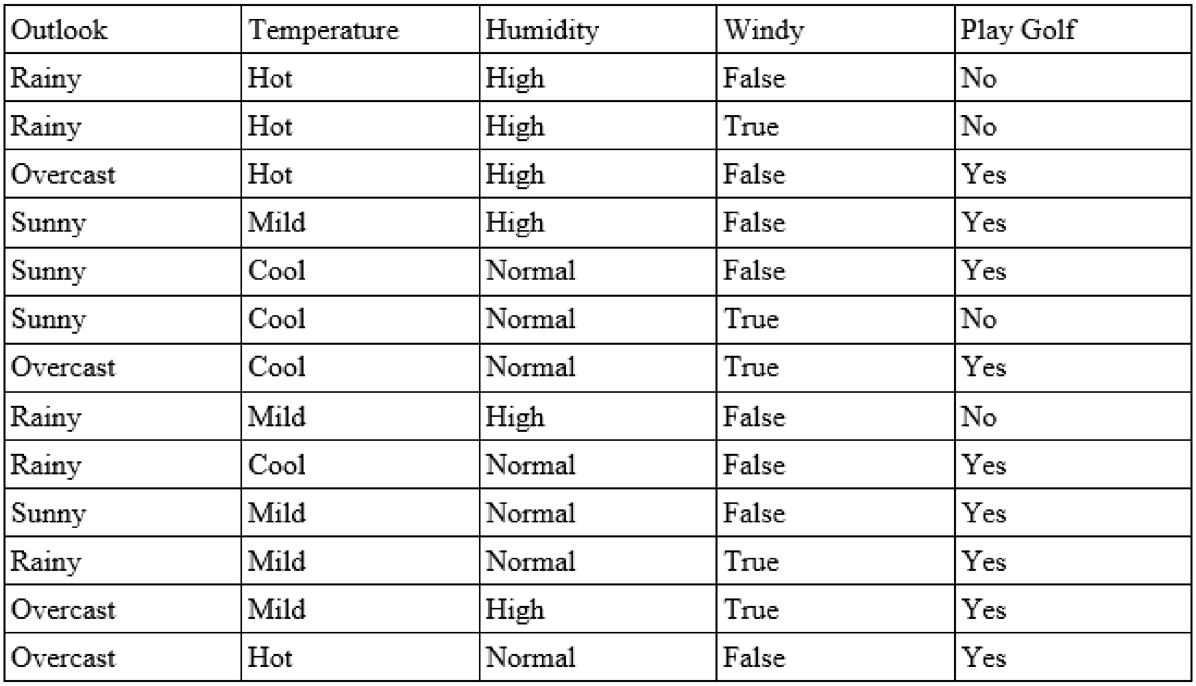
EXAMPLE:

Firstly, the aim is to calculate posterior probability.

We can do that by first, constructing a frequency table which will include each attribute for the target.

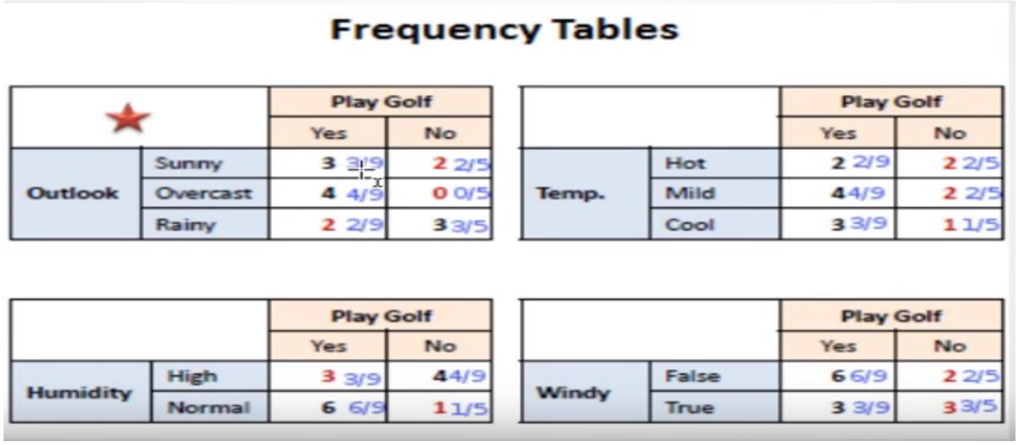
Then, we transform the freq. tables to likelihood tables and finally we use the Naïve Bayesian equation to calculate the posterior probability for each and every class.

The class with the highest posterior probability in the outcome of prediction.



Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

Here we have four categorical attributes and we have a class. Here we calculate the probability of a class that is the prior probability and then we build the frequency table as in OneR classifier and from each frequency table we extract different probabilities:



Frequency Tables

★		Play Golf	
		Yes	No
Outlook	Sunny	3 3/9	2 2/5
	Overcast	4 4/9	0 0/5
	Rainy	2 2/9	3 3/5

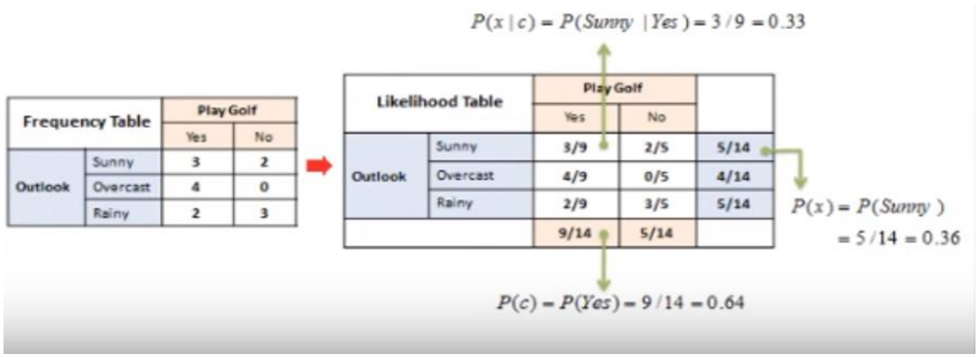
		Play Golf	
		Yes	No
Temp.	Hot	2 2/9	2 2/5
	Mild	4 4/9	2 2/5
	Cool	3 3/9	1 1/5

		Play Golf	
		Yes	No
Humidity	High	3 3/9	4 4/9
	Normal	6 6/9	1 1/5

		Play Golf	
		Yes	No
Windy	False	6 6/9	2 2/5
	True	3 3/9	3 3/5

Figure 2.6

Here probability of ‘sunny’ given it’s a “yes” is 3/9, similarly we calculate all the other probabilities. Now that we have taken out the probabilities from the frequency table, we can list them in a table shown below:



$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$

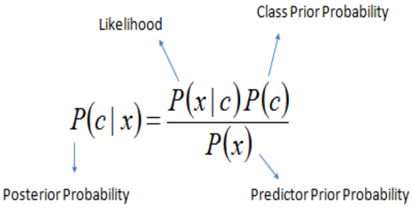
Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$P(x) = P(Sunny) = 5/14 = 0.36$

$P(c) = P(Yes) = 9/14 = 0.64$

Now as mentioned above we have the all the probabilities we need to calculate the probability of the class using the formula.



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

If we take a random input for example:

Outlook=Rainy

Temperature=Mild

Humidity=Normal

Windy=True

Likelihood of Yes=P(Outlook=Rainy|Yes)*P(Temperature=Mild|Yes)*P(Humidity = Normal|Yes)*P(Windy=True|Yes) =

2/9 * 4/9 * 6/9 * 9/14 = 0.0141

Likelihood of No=P(Outlook=Rainy|No)*P(Temperature=Mild|Yes)*P(Humidity=Normal|Yes)*P(Windy=True|Yes) =

3/5 * 2/5 * 1/5 * 3/5 * 5/14 = 0.0102

Now we normalize:

P(Yes)=0.014109347/(0.014109347+0.010285714) = 0.578368999

P(No)=0.010285714/(0.014109347+0.010285714) = 0.421631001

The Zero-Frequency Problem:

When the value of an attribute (Outlook=Overcast) doesn’t occur with every value of the (Play Golf=no) then this problem occurs.

To solve this problem we add 1 to all counts.

Decision Tree:

Decision tree aims at building the classification or regression model in the form of a simple tree structure.

It does so by breaking down a dataset into smaller and smaller subsets and at the same time incrementally developing an associated decision tree.

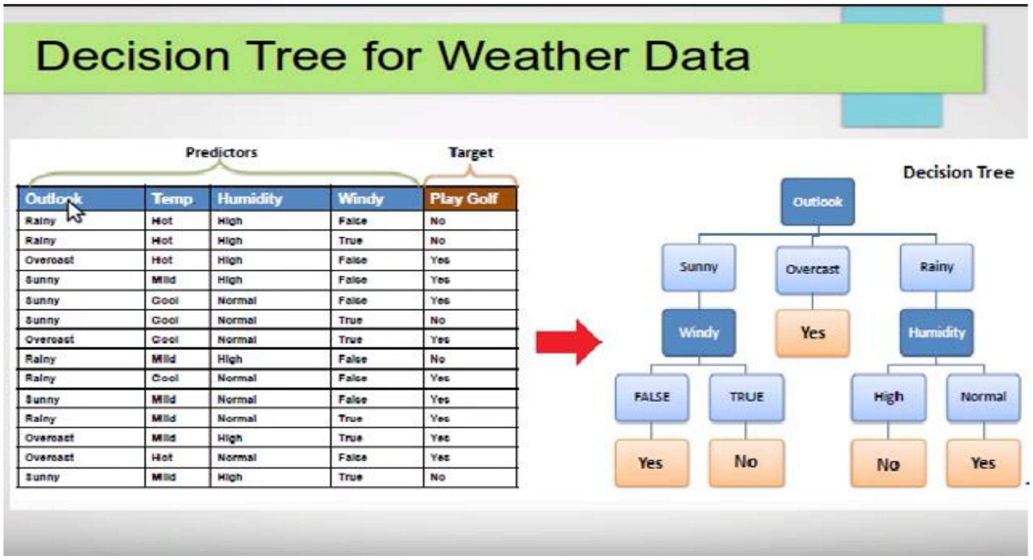
The final outcome is a tree with leaf nodes and decision nodes.

-A decision node(e.g. Outlook) consists of two or more branches .

-Leaf node (e.g., Play) depicts a classification or decision.

The topmost decision node which is related to the best predictor in a tree called root node.

Decision trees can handle both categorical and numerical data.



Decision Tree for Weather Data

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Figure 2.7

Taking the same example, we have three attributes and one target class. A decision tree when made from the data given above might look as shown in the figure above in which the best way to branch Outlook is in three categories namely-Sunny, Overcast and Rainy and similar is the case with Windy and Humidity. In the leaf nodes we keep elements of the target class that is "yes" or "no".

How It Works:

The main algorithm used for building decision trees which is called ID3 by J.R. Quinlann which employs a top-down ,greedy search through the space of possible branches with no backtracking.

Entropy and Information Gain are used by ID3 which helps in the construction of decision tree.

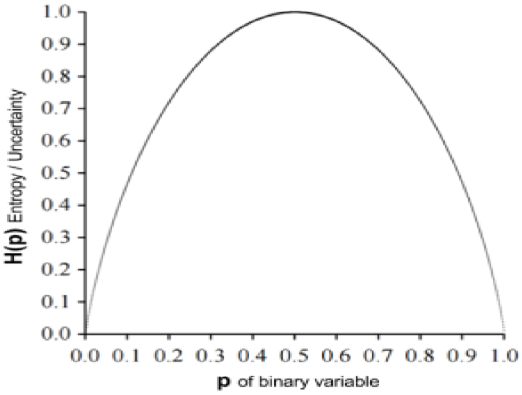
Entropy

Entropy is the measure of uncertainty and the value of Entropy is computed for two or three classes or categories and it is done by multiplying the probability of each category by the log to the base two of the value of that probability and summing that value of all the classes.

We build a decision tree top to down from a root node and we involve partitioning the data into subsets that consists of instances with similar values (homogeneous).

ID3 algorithm calculates the homogeneity of the sample using entropy.

If the sample is a complete homogeneous one the entropy is 0 and if the sample is an equally divided one, it has entropy of 1.

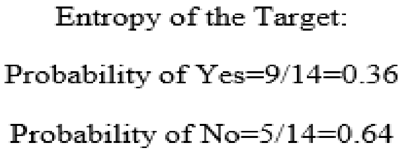


Compute Two Types Of Entropy:

Two types with the use of frequency tables are calculated in the process of building decision tree which are as follows:

Calculation of entropy using the frequency table of one attribute (Entropy of the Target) is as follows:

$$E(S)=\sum_{i=1}^{c} -pi \ \log_2 pi$$

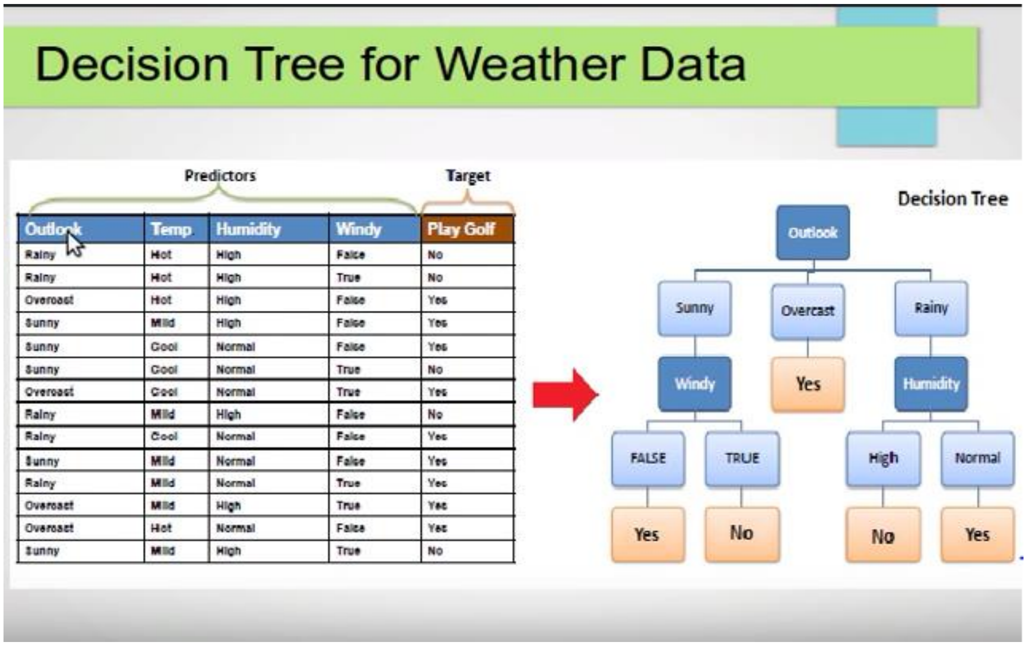


Entropy of the Target:

Probability of Yes=9/14=0.36

Probability of No=5/14=0.64

The minus sign is used to make the value of E(S) positive because the value of pi is between 0 and 1 and the value of log between 0 and 1 is always negative. So to nullify that negative sign there is a need of another negative sign in the formula. If we look at the previous figure, we notice that the entropy at the middle of the graph is highest that is 1. If things are equally divided then the entropy is 1 and if things are homogeneous then the entropy is 0.



Decision Tree for Weather Data

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Figure 2.8

Now here we will compute the entropy of play golf that is our target class . The calculation of the entropy is shown in the above figure.

Entropy using the frequency table of two attributes:

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

Now we focus on the frequency table for the outlook and notice the counts , we built the frequency table against the class values and now we are only concerned about the rows and not the columns and these must sum up to the number of instances. Now we compute the entropy for the target when we split by outlook and we do that multiplying the probability of that category by the entropy of that category as shown in the figure above.

Information Gain:

The information gain works on the basis of the decrease in entropy after a dataset is split on an attribute.

Gain(T,X) = Entropy (T) - Entropy(T,X)

Construction of a decision tree is to find the attribute that returns the highest information gain (i.e., the most homogeneous branches).

LINEAR DISCRIMINANT ANALYSIS

The classification method known as Linear Discriminant Analysis (LDA) is originally developed by R.A. Fisher in 1936.

In this , we reduce the number of variables or data preserving as much of the class discrimination. Having too many features does not means being a good model

It’s simplicity, mathematical robustness often produces models with accuracy comparable to more complex methods.

ALGORITHM:

LDA focuses on the concept of a linear combination of variables being searched (predictors) that best separate two classes (targets).

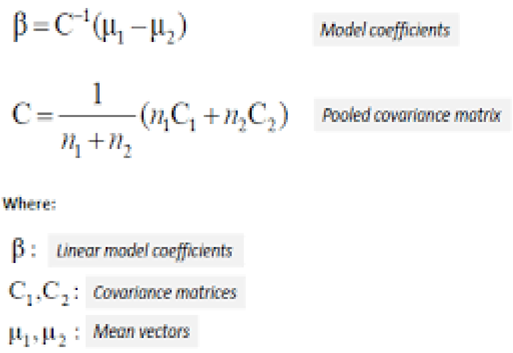
For the notion of separability to be captured, the following score function was defined by Fisher:

$$Z = \beta 1x1 + \beta 2x2 + \cdots + \beta dxd$$

$$\mathrm{S}(\beta)=\frac{\beta^T \mu 1-\beta^T \mu 2}{\beta^T C\beta}$$

$$\mathrm{S}(\beta)=\frac{\overline{Z1}-\overline{Z2}}{Varience\ of\ Z\ withingroups}$$

The problem is the estimation of the linear coefficients to maximize the score which can be solved by the following equations, given the score function:



$$\beta = C^{-1}(\mu_1 - \mu_2)$$ Model coefficients

$$C = \frac{1}{n_1 + n_2}(n_1 C_1 + n_2 C_2)$$ Pooled covariance matrix

Where:

β : Linear model coefficients

C_1, C_2 : Covariance matrices

μ_1, μ_2 : Mean vectors

The effectiveness of the discrimination can be accessed by calculating the Mahalanobis distance between the pair of groups.

If distance is greater than 3, it implies that the two averages have a difference of standard deviations more than 3.

It means that the overlap (probability of misclassification) is quite small.

Finally, the classification of a new point takes place when it is projected into the maximally separating direction and classified as C1 if:

$$\beta^T \left[x - \left[\frac{\mu 1 + \mu 2}{2} \right] \right] > \log \frac{p(c1)}{p(c2)}$$

LOGISTIC REGRESSION

To understand Logistic Regression, we should understand the idea of Linear Regression. We should know about the idea of best fit line. Basically, we have a predictor or variable. Logistic Regression focuses on prediction of the probability of an outcome that can be bi-valued (i.e. A dichotomy). It is basically used for binary classifications. It is used to answer questions.

The basis of prediction is the usage of 1 or many predictors (numerical and categorical).

There are two reasons why Linear Regression is not used for prediction of binary variables and those are as follows:

The predicted values by Linear Regression will be outside the expected range (e.g. Predicting probabilities outside the range 0 to 1)

Since only one of the two possible values can be taken by the dichotomous experiments for each experiment, the distribution of the residuals about the predicted line will not be normal.

A logistic curve is produced by Logistic Regression which has its limits between values 0 and 1.

Logistic and linear regression are similar to one another, but the construction of the curve is done using the natural logarithm of the “odds” of the target variable, rather than the probability.

As a matter of fact, the predictors do not have to have equal variance or do not have to be normally distributed in each group.

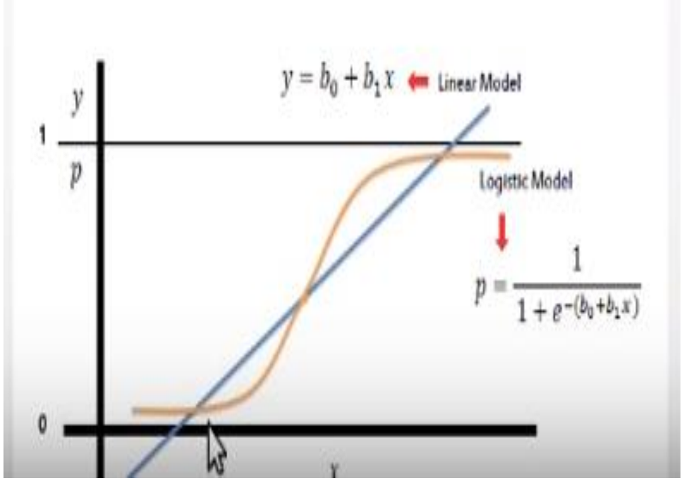


Figure 2.9

In the logistic regression the curve id moved to the left by constant (b0) and the slope (b1) which shifts the curve to the right defines the steepness of the curve.

The equation of the Logistic Regression, by simple transformation can be written as the ratio:

$$\frac{p}{1-p} = \exp\left(b_0 + b_1 x\right)$$

In the end, by taking the log of both sides, equation can be written in terms of log-odds (logit) which is a linear function of the predictors :

$$ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x$$

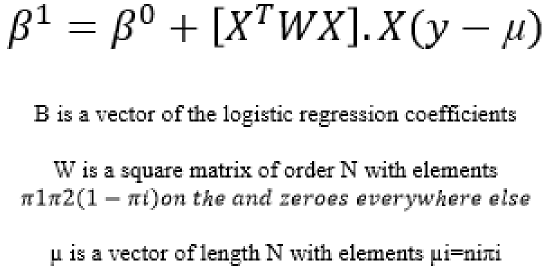
The coefficient (b1) is the amount the logita (logg-odds) changes with a one unit change in x.

As mentioned above, any number of numerical or categorical values can be handled by the logistic Regression.

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p)}}$$

Linear and Logistic Regression

Linear and logistic regression have several theories between them. Just as ordinary the least square regression method is used to calculate and estimate coefficients for the best fit line in linear regression, maximum likelihood estimation(MLE) is used by logistic regression to get the model coefficients that relate predictors to the targets.
After this the estimation of initial function takes place, the process is repeated until LL(Log Likelihood) remains constant.



$$\beta^1 = \beta^0 + [X^T W X].X(y - \mu)$$

B is a vector of the logistic regression coefficients

W is a square matrix of order N with elements
$\pi 1 \pi 2(1 - \pi i)$*on the and zeroes everywhere else*

μ is a vector of length N with elements μi=niπi

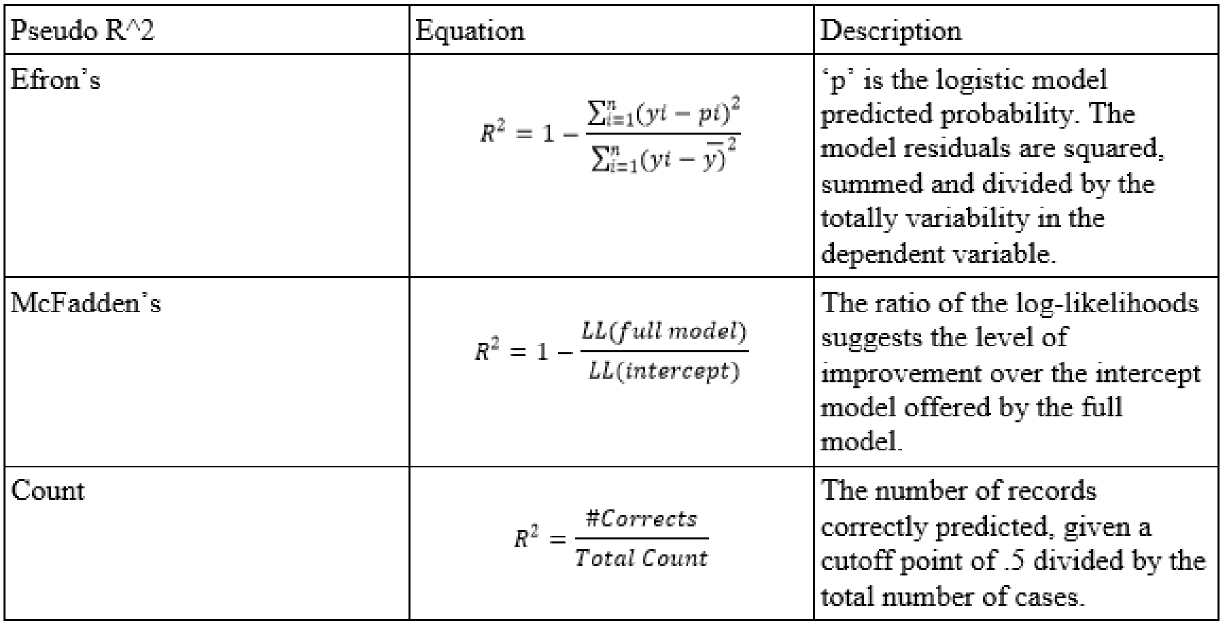
Pseudo R-Squared

This is also present to basically indicate the adequacies of the regressions model.The likelihood ratio test is actually a test of the present significance of the main differences between the actual likelihood ratio for the basic baseline model when minus the actual likelihoods ratio for all the baselined model minus the likelihoods ratio for the reduced model.

This basic difference is actually called “model chi-square”.

This statistical significance is actually tested while using the actual Walds test of each of the coefficient (b) in the model(i.e., predictors contributions).

There are actually several measures which intend to mimic mostly the R-squared analysis in evaluating the goodness of fit of logistics models but what they can not be interpreted as the one would actually interact with an R-squared and the different pseudo R-squared can arrive at a very different values.
Here we can discuss the three pseudo R-Squared measures .



Pseudo R^2	Equation	Description
Efron's	$R^2 = 1 - \frac{\sum_{i=1}^{n}(yi - pi)^2}{\sum_{i=1}^{n}(yi - \bar{y})^2}$	'p' is the logistic model predicted probability. The model residuals are squared, summed and divided by the totally variability in the dependent variable.
McFadden's	$R^2 = 1 - \frac{LL(full\ model)}{LL(intercept)}$	The ratio of the log-likelihoods suggests the level of improvement over the intercept model offered by the full model.
Count	$R^2 = \frac{\#Corrects}{Total\ Count}$	The number of records correctly predicted, given a cutoff point of .5 divided by the total number of cases.

Likelihood Ratio Test

The likelihoods ratio test is used to provide the means which is used for comparing In the likelihood in case of of the data which is done under one of the models(e.g. full model) .It is done against all the likelihood of this data under another data which is a more restricted model (e.g. intercept model)

where 'p' is none other than the logistic model which is predicted probability.

Wald Test

The evaluation of the this test is the sigificance statistically of each of the coefficient (b) in just the model is accomplished using the Wald test.

In this case W is none other than the Wald's statistic with all the normalized distribution (like the Z-test) , b is just the coefficient and SE in this case is the Standard Error.

The Wald statistic with the chi -square distributions is actually yielded by all the squaring of the W value.

Predictor Contributions:

Wald's test is just usually and basically used to assess all the significant prediction of each of the predictor.
One more indicator of the contribution of a predictor is basically exp(b) or it is odds-ratio of the coefficient which is nothing else than amount of just the logit (logg-odds) changes , with a one unit of the change in the predictor (x).

K Nearest Neighbours

K nearest neighbours is an algorithm which is used to store all the cases and classifiers available new cases which are particularly based on the similarity measure, for example distance functions.

KNN has all been used in the statistical estimations and in the pattern recognition already in the beginnings of the 1970's.

Algorithm:

Near neighbors come together to vote which help in classification with the case which is being assigned to one of the class which is the most common one among the K members and they measure the distance using distance function

If K=1 , then this particular case simply assigns to the class of its own nearest of the neighbor.

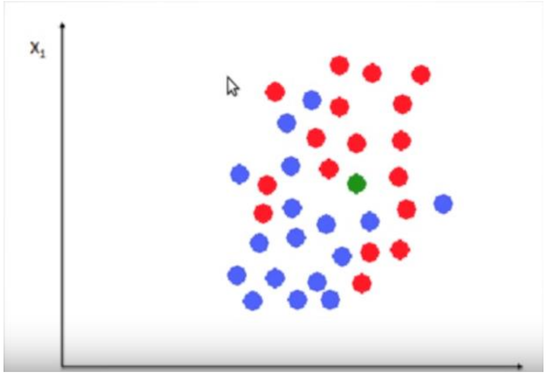
Example:

Let us assume we have a new green member in the surroundings . We have to predict if the member is male or female. We measure the distance between it and the nearest neighbor . Let red members be male while blue memberas be female .

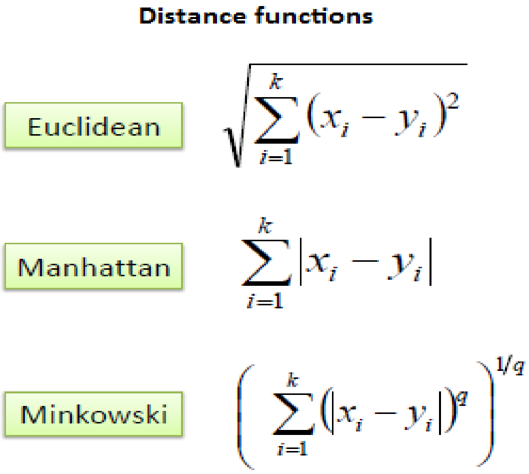
K should be odd for calculating the gender

If K=1 , the nearest member is male. So green is male.

If K=5 , 3 nearest neighbors are males while 2 are females , so we can say as the majority is of males , it is a male member .



DISTANCE MEASURED BY CONSTANT VARIABLES



Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

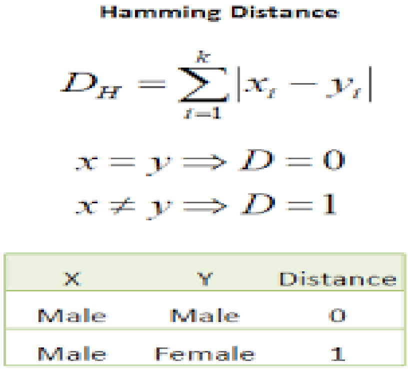
Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Categorical Variables

In the instance of the categorical variables the Hammings distance should be used.

The issue of the standardization of the nummerical variables is brought between the values 0 and 1 when there is nothing but the numerical and the categorical variables mixed within the dataset.



Hamming Distance

$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

$$x = y \Rightarrow D = 0$$
$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

How many neighbors?

The optimal value can be chosen for the value of K and the best way to do it is to inspect the data.

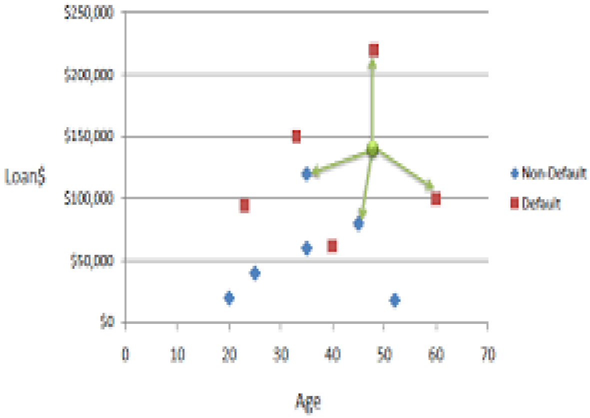
Precision is proportional to value of K as there is reduction in the noise but it is without any guarantee.

Cross- Validation is an another way to determine retrospectively a good K value by usage of an independent dataset which is done to validate the value of K.

A optimal value of K for the most of the datasets is mostly between the value 3-10 according to past research which is responsible for producing much better results than 1NN

Example:

Consider the following data concerning creditdefault. Age and Loan are two numerical variables (predictors) and Default is the target .

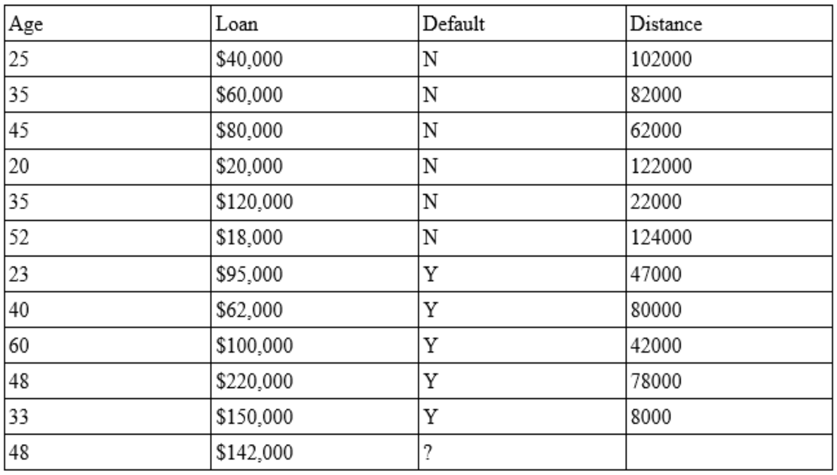


Graph 2.3

the training set can be used to classify the unknown cases(Age is 48 and Loan is $1,42,000) using the basic Euclidean distance.

If the K=1 then the nearest neighbour is obviously the last one case in the training set with Default=Y.

D=Sqrt [(48-33)^2 +(142000-150000)^2] =8000.01 >> Default =Y



Age	Loan	Default	Distance
25	$40,000	N	102000
35	$60,000	N	82000
45	$80,000	N	62000
20	$20,000	N	122000
35	$120,000	N	22000
52	$18,000	N	124000
23	$95,000	Y	47000
40	$62,000	Y	80000
60	$100,000	Y	42000
48	$220,000	Y	78000
33	$150,000	Y	8000
48	$142,000	?	

$$D = \sqrt{(x1 - y1)^2 + (x2 - y2)^2}$$

With K=3, there are two Default values of =Y and the one Default =N out of 3 closest neighbors.
The predictied Unknown case is the Default value of Y.

Standardized Distance:

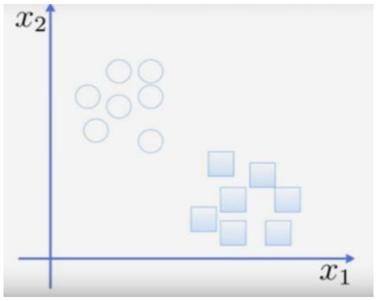
One of the major drawbacks is in calculating and measure the distance measures directly from none other than the training set as it is in the case where there are variables having different measurement of scales or there is the mixture of all the numerical and/or categorical variables.

Like for example , if one of the variable is basically based on the annual income which is in dollars, and the other one is based basically on the age of 9in years then the income will undoubtedly have a higher impact on the distance which is calculated.
One solution can be the standardizing of the training sets.

LINEAR SVM

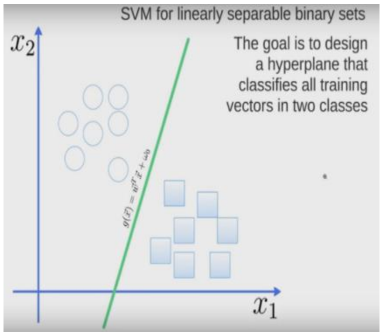
The aim of linear SVM is to designe a hyperplane for classifying all training vectors in two separate classes. If we have two different hyperplanes which can classify correctly all of the instances a class has. If we have to choose between them, the best one will the one that leaves the maximum amount of margin from both the instance of classes.

The margin is nothing but the distance between the so-called hyperplane and of the closest of the elements of the class from the hyperplane.



Graph 2.4

In case of the red hyperplane, the margin we have is represented as Z1 whereas in case of the green hyperplane, the margin is represented as Z2. Here the value of Z2>Z1. So the margin is higher in the case of the green margin. This implies the best choice will be the green hyperplane.



Graph 2.5

The green hyperplane is represented by the following equation:

$$g(\overrightarrow{x}) = \omega^{-\tau} + \omega_0$$
$$\mathrm{g}(\overrightarrow{x}) \geq 1 \quad \forall\, \vec{x} \in \text{class } 1$$
$$\mathrm{g}(\vec{x}) <= -1 \quad \forall\, \vec{x} \in \text{class } 2$$

We can say that the distance from the closest elements will be at least 1 (the modulus is 1) and from the geometry we know that the distance between two points in a hyperplane can be easily computed by the following equation:

$$z = \frac{|g(\vec{x})|}{||\vec{\omega}||} = \frac{1}{||\vec{\omega}||}$$

So the total margin which corresponds to the distance between the classes and their specific hyperplane is computed by:

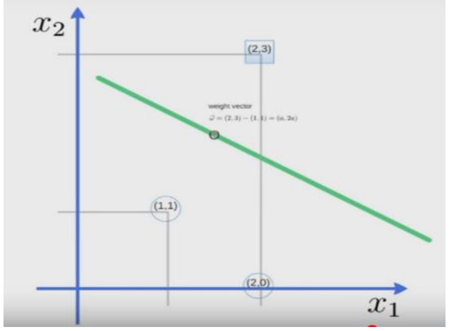
$$\frac{1}{||\vec{\omega}||} + \frac{1}{||\vec{\omega}||} = \frac{2}{||\vec{\omega}||}$$

By minimizing the term on the right side we can maximize the separability which means if this factor would be minimized, we will just have one of the biggest margin which will separate the two classes.
$\vec{\omega}$ can be minimized by a number of non linear optimization task solved by the Karushastr-Kuhn-Tuckers(KKT) conditions, using Langrange multipliers $л_i$.
The mail equation states that:

$$\vec{\omega} = \sum_{i=0}^{N} л_i y_i \vec{x}_i$$

$$\sum_{i=0}^{N} л_i y_i = 0$$

When we solve these equations trying to minimize the $\vec{\omega}$, we will maximize the margin between the two classes. That means we will maximize the separability between the two classes.
EXAMPLE:
Suppose that we have these two features like X1 and X2 and all that we have the three values mentioned in the figure and we have to find the best hyperplane that will divide these two classes:



So we can see clearly from the above graph that the best division line will be a parallel line that connect the two values. So we can define the following weight vector:

$$\vec{\omega} = (2{,}3) - (1{,}1) = (a, 2a)$$

Now we can solve this weight vector and create the hyperplane equation using this weight vector as follows:

Weight vector $\vec{\omega} = (a, 2a)$
a+2a+$\omega_0 = -1$ using point(1,1)
2a+6a+$\omega_0 = 1$ using point(2,3)

$$\cdots$$

$$\omega_0 = 1 - 8a \qquad 3a + 1 - 8a = -1$$

$$.$$

$$. \qquad\qquad 5a = 2$$

$$. \qquad\qquad a = \tfrac{2}{5}$$

$$\omega_0 = 1 - 8\frac{2}{5} = \frac{5-16}{5}$$

$$\omega_0 = \frac{11}{5}$$

$$\cdots$$

$$\vec{\omega} = \left(\frac{2}{5}, \frac{4}{5}\right)$$

$$g(\vec{x}) = \frac{2}{5}x1 + \frac{4}{5}x2 - \frac{11}{5}$$

$$g(\vec{x}) = x_1 + 2x_2 - 5.5$$

Hence now we have the equations of the hyperplane which will divide the two classes with maximum separability.

RANDOM FOREST

Random Forest (Random Decision Forest) are nothing but an ensemble method of learning which is used for regressionor classification and to perform other tasks that usually operate with the multitude of the decision trees classification at the time of training and it also helps in showing the output of the class that basically classifies the mode of all the classes or mean prediction(regression) of the individual trees.It is used to correct for decision trees' habit of the overfitting to the training sets.

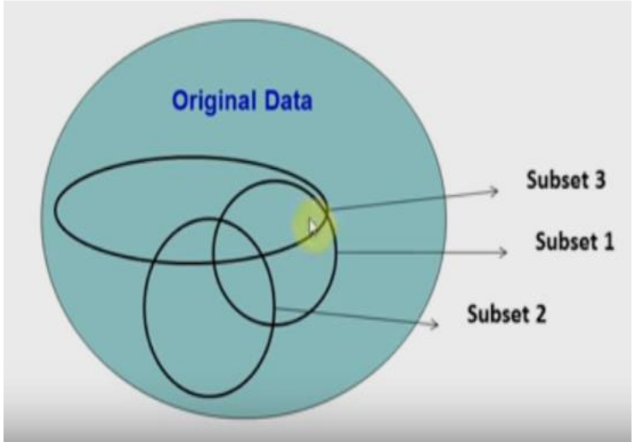
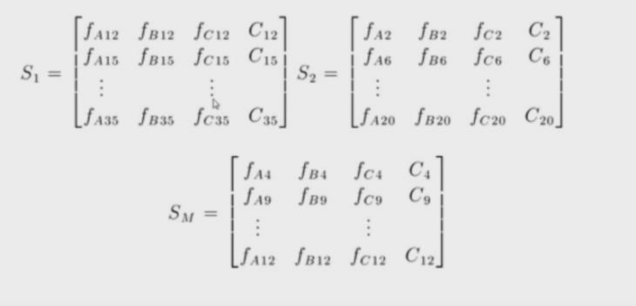


Figure 2.10

ALGORITHM USED:

Decision trees basically are used widely for tasks involving machine learning. This tree learning method has an advantage of meeting of the requirements which serve as the off-shell procedure of the data mining because this method is not variant in case of scaling or for performing various and many other transformations of values of feature, which is robust in the inclusion of many not relevant features and is used to produce models which are inspectable. However, they are not always true or accurate.

CREATING RANDOM SUBSETS:



$$S_1 = \begin{bmatrix} f_{A12} & f_{B12} & f_{C12} & C_{12} \\ f_{A15} & f_{B15} & f_{C15} & C_{15} \\ \vdots & & \vdots & \\ f_{A35} & f_{B35} & f_{C35} & C_{35} \end{bmatrix} S_2 = \begin{bmatrix} f_{A2} & f_{B2} & f_{C2} & C_2 \\ f_{A6} & f_{B6} & f_{C6} & C_6 \\ \vdots & & \vdots & \\ f_{A20} & f_{B20} & f_{C20} & C_{20} \end{bmatrix}$$

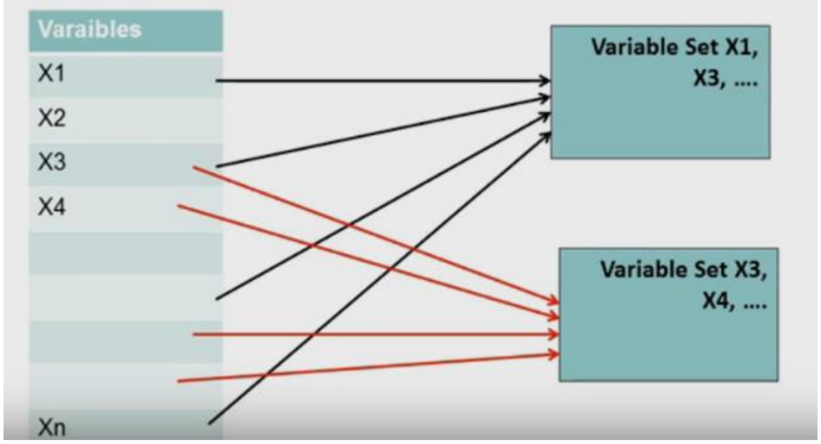
$$S_M = \begin{bmatrix} f_{A4} & f_{B4} & f_{C4} & C_4 \\ f_{A9} & f_{B9} & f_{C9} & C_9 \\ \vdots & & \vdots & \\ f_{A12} & f_{B12} & f_{C12} & C_{12} \end{bmatrix}$$

Particularly, trees do grow deeply and the irregular patterns tend to be in place. The training sets are overfitted as they have very low bias but it has a very high variance. Multiple decision trees average which is a way for random forests and are trained on the various different parts of exactly same various different parts and the goal is the reduce the variance.

BASIC PRINCIPLES IN RANDOM FOREST METHOD:

Based on the selection of data and variables randomly ,lots of decision tress are developed.

Following is how the variables are selected randomly:



Relationship to nearest neighbors

Random forest and k-nearest neighbor algo can be related. This was discovered by Lin and Jeon in 2002. Both can together be seen as weighted neighborhoods schemes. The new points are predicted by the models which are mainly built from training set.This is done by looking for the neighborhood of the points which is formalized by the W’ (Weight Function).

Now the *i*'th training point has the non-negative weight which actually related to the new point of the same tree whicbh is x’. For any of the particular *x′*, sum of the weights for points should actually sum to one.

These weight functions are the following:

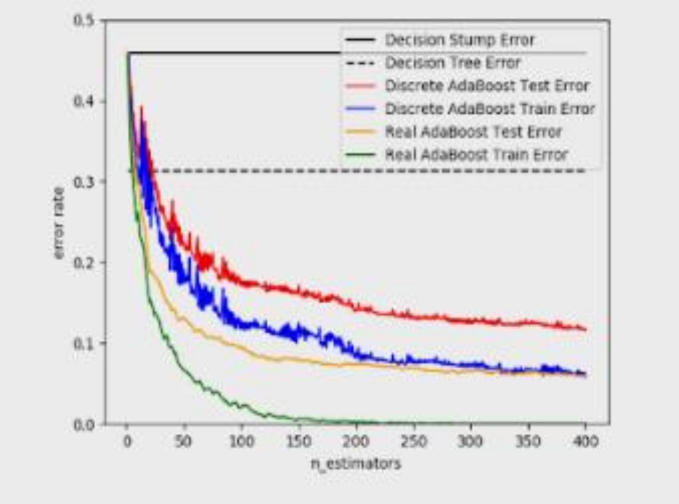
- In *k*-NN, the weights will be W(x_i ,x’)=1/k’ if x_i is one of the *k* points is the closest to *x′ and* zero if not.
- In a tree, W(x_i,x’)=1/k’ if x_i is actually either one of the *k′* points are in the same leaf as *x′ or* zero otherwise.

$$\hat{y} = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n} W_j(x_i, x')\, y_i = \sum_{i=1}^{n}\left(\frac{1}{m}\sum_{j=1}^{m} W_j(x_i, x')\right) y_i .$$

Since the forest is used to average the so-called predictions of like a set of m trees ,each having their own functions of individual weight

This all shows that the forest as a whole is nothing but again the weighted neighborhood schema which has the weights that particularly do average those of the trees which are individual. The neighbors of x' in this schema are those points which share the same leaf in any of the trees.

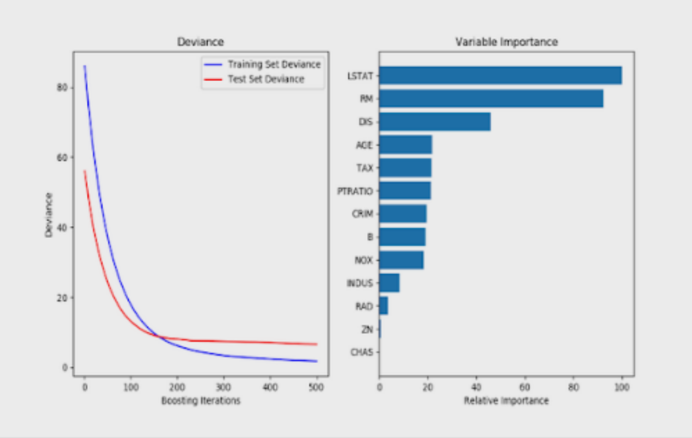
In this schema,the structure of the trees highly influence the neighborhood of x' and thus it depends on nothing but the structure of nothing but the training set. Lin as well as Jeon show that the local importance of each feature gets adapts the shape of the neighborhood which is used by random forest.



Graph 2.6

Rationale

It is nothing but the combination of learning models which is used to increase the accuracy of classification.



Gradient Tree Boosting

Gradient Boosted Regression Trees (GBRT) or Gradient Tree Boosting is basically a general way to boost the arbitrary differentail loss functions. It is an accurate and is effective method which is off-shell which is starting to be used for either regression or classification problems.

They are actually used in many areas which includes Web search rankings or ecology.

GBRT has its disadvantages:

- The way it handlles the data of mixed type naturally.(heterogeneous features)
- Power of prediction.
- Robustness to all the outliers in the output space which is done through robust loss functions.

GBRT has its disadvantages:

- Scalability which happens because of sequential nature of boosting and nonetheless it can hardly be parallelized.

CHAPTER-3

SYSTEM DEVELOPMENT

HARDWARE AND SOFTWARE REQUIREMENTS:

-Hardware Requirements:

12-24 1-4TB hard disks in the JBOD (Just a Bunch Of Disks) configuration

2 quad-/hex-/octo-core CPUs, running at least 2-2.5GHz

64-512 GB of RAM

10Gigabit Ethernet

-Software Requirements :

Oracle Virtual Box

Ubuntu 16.04

Hadoop 2.6.2

CHAPTER-4

PERFORMANCE ANALYSIS

Comparing the performance in terms of time and accuracy of different classifiers is very important consistently.

We can compare multiple classifier algorithms in python with scikit-learn and also add our own algorithm to be compared.

Choosing The Best Classifier Algorithm:

Looking at the data from different perspectives, we can compare and contrast different classifier algorithms and choose the best amongst them.

There are a lot of ways to estimate the accuracy of the algorithms in order to narrow it down to one or two to make the final choice.

To do this we use different perspectives to look at the data and select a particular model such as mean accuracy, standard deviation, variance etc. to make sure we select the correct algorithm to classify our data.

Comparing Classifier Algorithms:

Comparison between algorithms can only be done by making sure that the algorithms are being evaluated by the same way on the same data.

Here we are taking a program to compare the following 6 algorithms:

-Logistic Regression

-Linear Discriminant Analysis

-K-Nearest Neighbors

-Naïve bayes

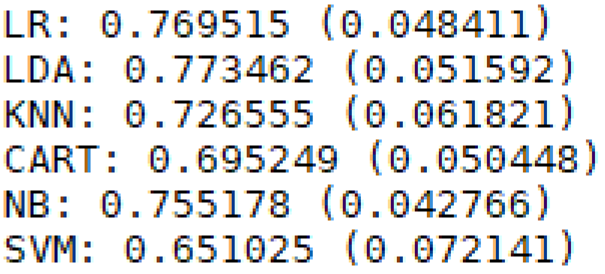
-Support Vector Machines(SVM)

-Decision Tree

Explanation:

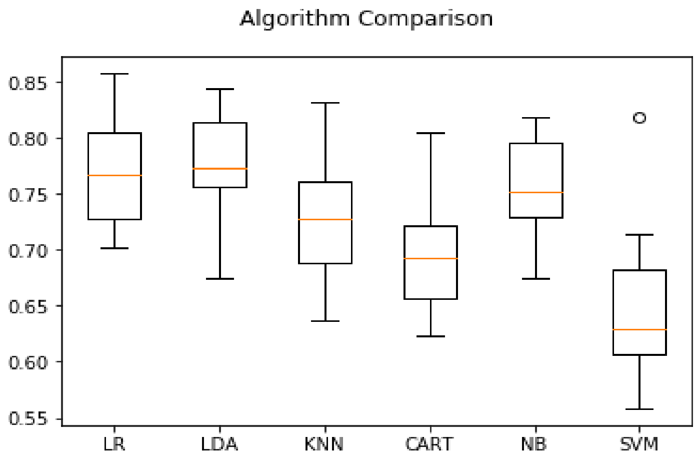
In the above code we are using pandas which is nothing but a library in scikit-learn for running python codes. We are importing the algorithms of all the above mentioned algorithms and passing the url “https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data" in order to process the data into these algorithms and so take out the time and accuracy of each algorithm to compare the contrasting results and choose the best algorithm.

Results:



```
LR: 0.769515 (0.048411)
LDA: 0.773462 (0.051592)
KNN: 0.726555 (0.061821)
CART: 0.695249 (0.050448)
NB: 0.755178 (0.042766)
SVM: 0.651025 (0.072141)
```

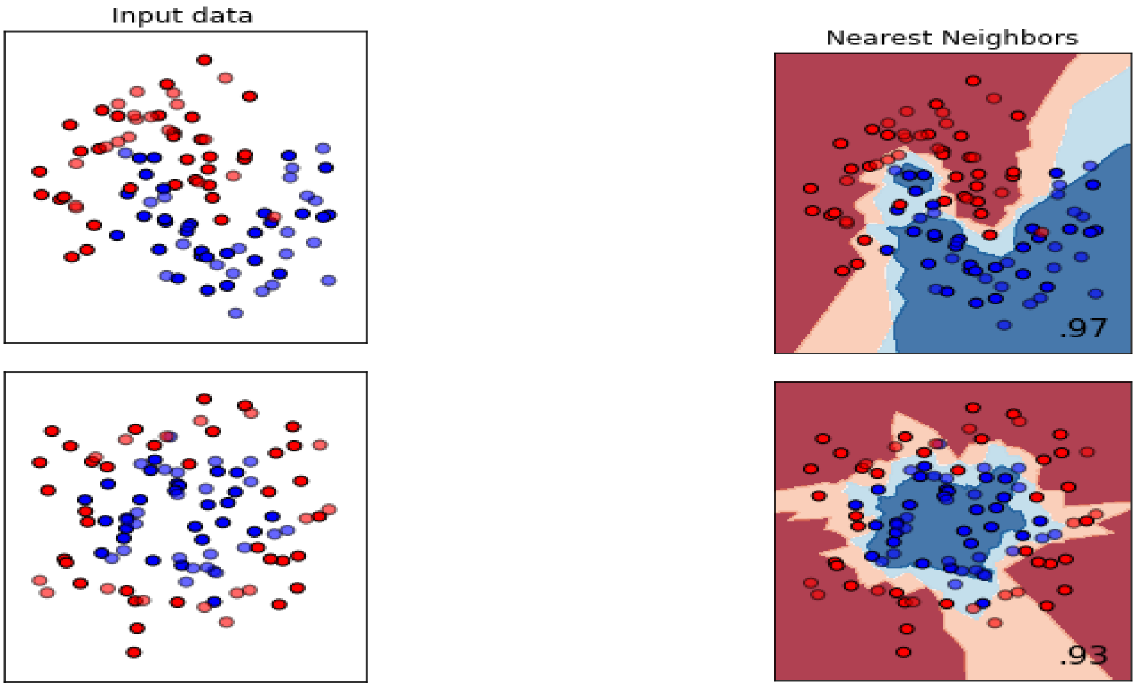
The above data shows the comparison of time and accuracy in different algorithms by means of which we can compare and contrast each algorithm and choose the most accurate one which takes the least time. We can also see the result on a graph shown below:

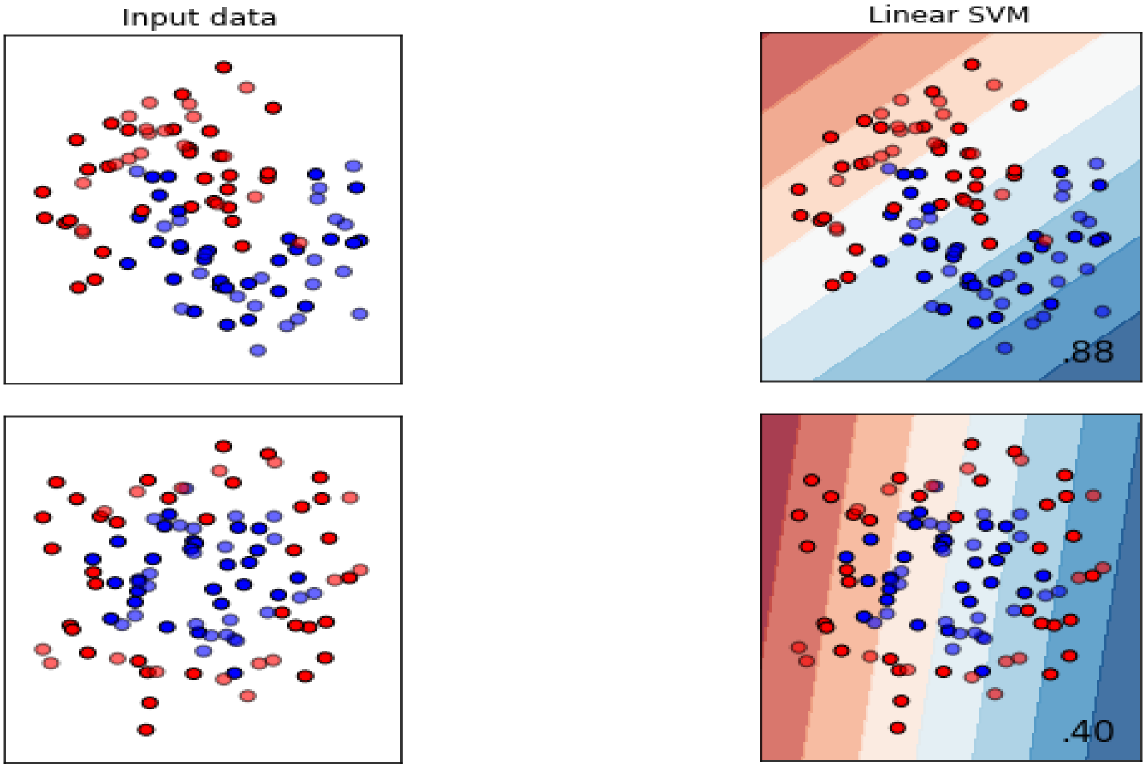


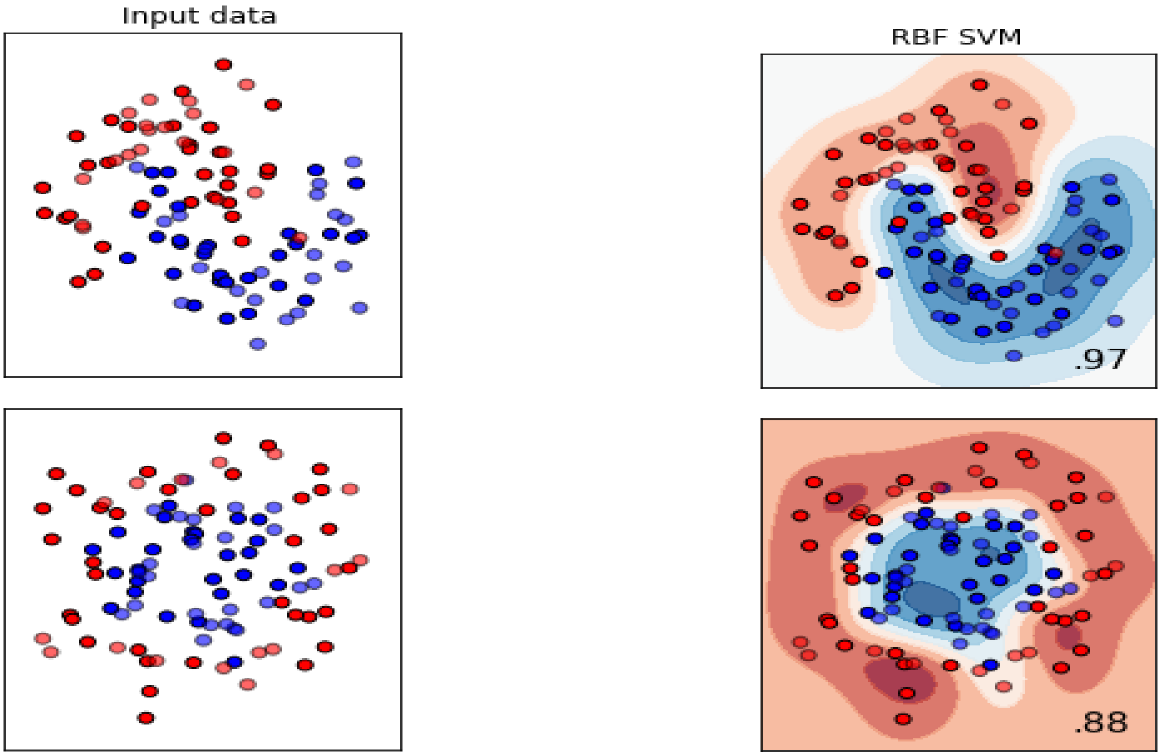
DIFFERENCE BETWEEN DATASETS OF DIFFERENT CLASSIFIERS

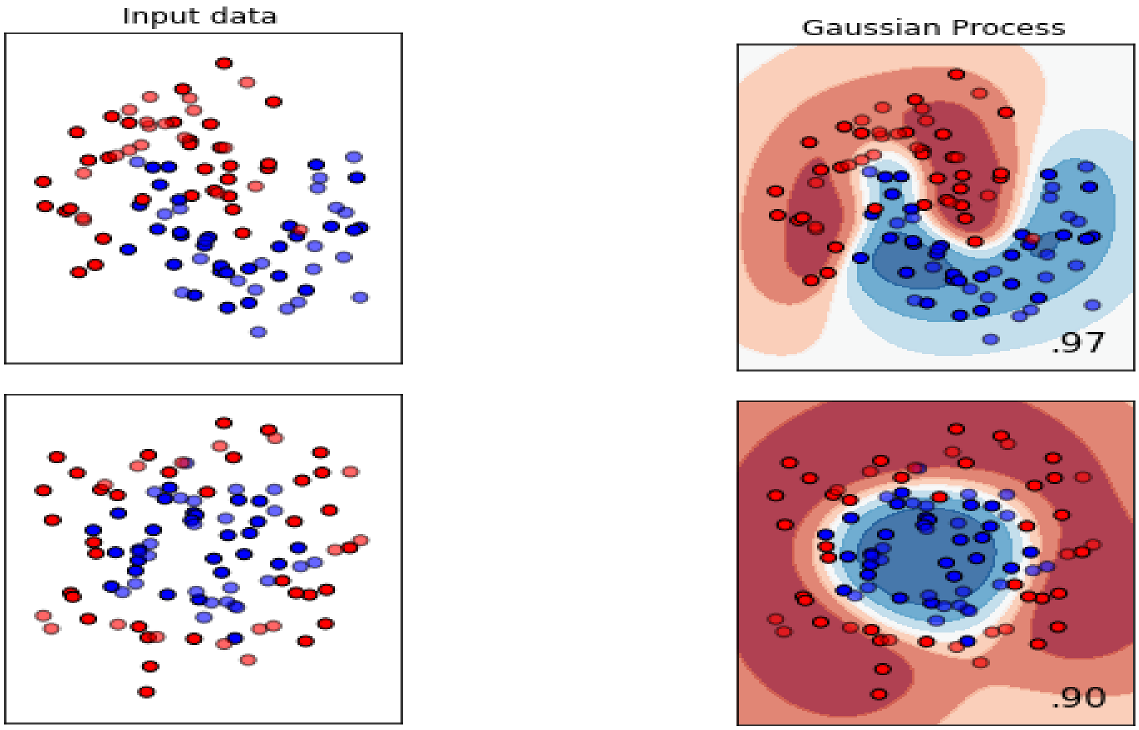
Several classifiers on synthetic datasets can be compared. It basically illustrates the varying nature of the decision boundaries of various different classifiers. A grainof salt is to be taken as these are not able to be implemented on to real datasets.
Well, to separate data in linear form and we can hence generalize in a better way simple classifiers such as Naïve Bayes and linear SVMs. They perform better than other classifiers.

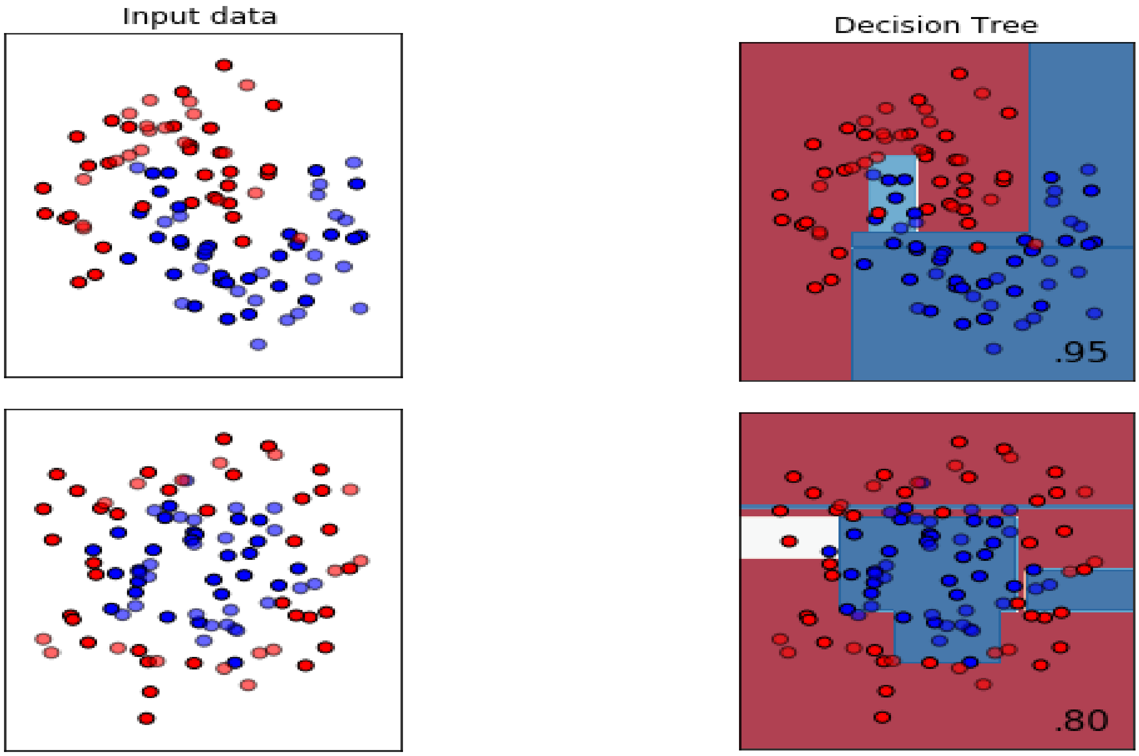
Following plots hence explain the training points in the solid car while the testing points are shown in semi-transparent.

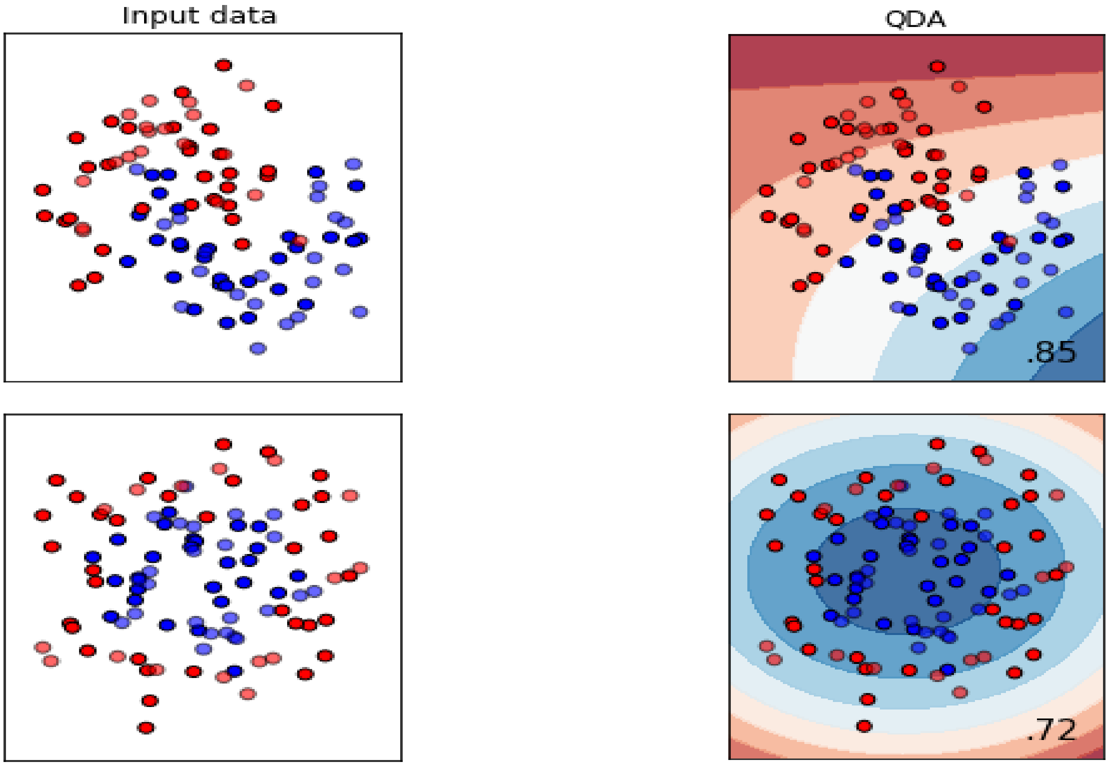



Input data
Linear SVM
.88
.40


Input data
RBF SVM
.97
.88


Input data
Gaussian Process
.97
.90


Input data
Decision Tree
.95
.80


Input data
QDA
.85
.72

CHAPTER-5
CONCLUSION AND FUTURE USE

Conclusion

In the end , we can conclude that there are various types of classifiers like decision tree, one R , zero R, Naive Bayes ,Linear Determinant Analysis, Logistic Regression and so on. These classifiers are used to classify the big data accourding to various parameters.

In Map Reduce algorithm, Mapper uses classifiers or predictor to map the data into smaller parts . It breaks the data into small bits by classifying into various forms. Then the reducer takes the already classified data and then it reduces it into desired form. We aim at achieving maximum efficiency using the required classifier and if possible there is a scope for creating a new classifier , we can develop our own classifier and can use it to find the minimum time and maximum efficiency.

Classifiers can be written in python , Java or R. Only Java programming was accepted earlier but with the update in various Map Reduced algorithms ,it is now possible to use python and R.With these algorithms , we can predict the execution time , execute the entropy of various classifiers and compare them. Comparison gives us the clear picture of which type of classifier can be used under which condition. We can take into account if there is any other scope for better classification and then apply it in Map Reduced algorithms for the future use.

Future Use

The future use of these type of algorithms will have a simple target of increasing efficiency, so that execution will be fast and big data will be analyzed faster and can be compared among themselves for best classification and work efficiently.This analysis will help in future as the main problem now a days is not the accumulation of large amount of data but analyzing that large amount of data, finding which data is relevant or not and organizing it into desired format.

We use hardware like ubuntu operating system and hadoop latest version for maximum accuracy.

REFERENCES:

Jefry Dean and Sanjay Ghemwat, MapReduce:A Flexible Data Processing Tool, Communications of the ACM, Volume 53, Issuse.1,January 2010, pp 72-77.

[2] Jefry Dean and Sanjay Ghemwat,.MapReduce: Simplified data processing on large clusters, Communications of the ACM, Volume 51 pp. 107–113, 2008

[3] Brad Brown, Michael Chui, and James Manyika, Are you ready for the era of „big data‟?,McInstitute, October 2011.

[4] M. Dhavapriya, N. Yasodha :Big Data Analytics: Challenges and Solutions Using Hadoop, Map Reduce and Big Table,October 2016.

[5] Katarina Grolinger1 , Michael Hayes1 , Wilson A. Higashino1,2, Alexandra L'Heureux1 David S. Allison1,3,4,5, Miriam A.M. Capretz1:Challenges for Map Reduce in Big Data,2014.

[6] Kevin P.Murphy:Naive Bayes classifiers, 2006.

[7] Kyuseok Shim, MapReduce Algorithms for Big Data Analysis, DNIS 2013, LNCS 7813, pp. 44–48, 2013.

[8] Raja.Appuswamy,ChristosGkantsidis,DushyanthNarayanan,Ori onHodson,AntonyRowstron, Nobody ever got fired for buying a cluster, Microsoft Research, Cambridge, UK, Technical Report,MSR-TR-2013-2

[9] Carlos Ordonez, Algorithms and Optimizations for Big Data Analytics: Cubes, Tech Talks,University of Houston, USA.